Amendment to the Claims:

The claims under examination in this application, including their current status and changes made in this paper, are respectfully presented.

1 (original). A method of operating a digital system having a processor and a memory, comprising the steps of:

executing a plurality of program tasks on the processor;

requesting a memory transaction responsive to a first one of the plurality of program tasks by providing a transaction address to be accessed and an identifier value indicative of the first program task;

detecting an error condition that prevents normal completion of the memory transaction; and

recovering from the error condition by using the identifier value provided with the memory transaction request to identify the first program task as the source of the memory transaction request.

2 (currently amended). The method of Claim ~~2~~ 1, wherein the identifier value comprises a task identifier value.

3 (original). The method according to Claim 1, wherein the identifier value comprises a resource identifier value to identify which resource of a plurality of resources is the source of the memory transaction request.

4 (original). The method according to Claim 1, further comprising the step of delaying the memory transaction request such that a second one of the plurality of program tasks is being executed when the error condition is detected.

5 (original). The method according to Claim 4, wherein the step of delaying the memory transaction request comprises queuing the identifier value along with the transaction address.

6 (original). The method according to Claim 4, wherein the step of delaying the memory transaction comprises keeping the identifier value coherent with an associated delayed memory access.

7 (original). The method according to Claim 4, wherein the memory transaction request resulted from a cache load or clean operation.

8 (original). The method of Claim 1, wherein the source of the memory transaction request is a direct memory access (DMA) engine.

9 (original). The method of Claim 1, wherein the source of the memory transaction request is a co-processor.

10 (original). The method according to Claim 1, wherein the step of recovering comprises suspending operation of just the first program task such that a remainder of the plurality of program tasks continue to be executed.

11 (original). A digital system, comprising:

a processor for executing program tasks, the processor having an address output port connected to a memory circuit, the address port operable to provide transaction addresses for memory transactions;

identifier circuitry connected to the processor for holding an identifier value indicative of a program task being executed;

abort circuitry connected to the identifier circuitry and to the address output port, the abort circuitry operable to store a fault address provided on the address output port and an identifier value corresponding to the fault address in response to a memory transaction error; and

wherein the processor is connected to the abort circuitry such that the processor is operable to read the stored fault address and the stored identifier value.

12 (original). The digital system of Claim 11, further comprising buffer circuitry connected between the address output port and the memory circuit, the buffer circuitry also

connected to receive identifier values from the identifier circuitry, the buffer circuitry also having outputs connected to the abort circuitry, wherein the buffer circuitry is operable to queue transaction addresses and corresponding identifier values and to provide them to the abort circuitry after a delay.

13 (original). The digital system according to Claim 12 being a personal digital assistant, further comprising:

a display, connected to the processor via a display adapter;

radio frequency (RF) circuitry connected to the processor; and

an aerial connected to the RF circuitry.